REMARKS

Claims 1 and 13 have been amended to correct typographical errors. Claims 1-35 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Declaration:

The Examiner asserts that a new declaration is required because the priority claim to provisional application 60/286,407 is incorrect per the Amendment filed on November 1, 2004, arguing that according to MPEP 602.01, the wording of a declaration cannot be amended. Applicant notes that no request was made to amend the originally submitted declaration; rather, in the previously filed Amendment, the Examiner was requested to disregard the incorrect priority claim. Applicant submits that a priority claim is not a required feature of the declaration according to 37 CFR 1.63. In fact, including a notation of a priority claim in a declaration is neither necessary nor sufficient to perfect such a claim. As noted in CFR 1.78(a)(2)(iii), a priority claim requires that a reference to the priority document be included in an application data sheet or in the first sentence of the specification following the title. Because a priority claim is not required to be listed in the declaration, and because such a listing would be insufficient to perfect the claim even if the claim were correct and intended, there exists no basis for concluding that inclusion of an unintended priority claim renders the declaration defective. Applicant notes that the declaration meets all requirements of 37 CFR 1.63. As such, Applicant submits that the declaration is correct with respect to its required features, and therefore no new declaration is necessary.

Objection to the Specification:

The Examiner objected to the specification due to several instances in which reference numerals did not agree with the corresponding drawing. The specification has

been amended as indicated above to correct the errors, and Applicant submits that this objection has been overcome.

Claim Objections:

The Examiner objected to claims 1 and 13 due to typographical errors. Claims 1 and 13 have been amended as indicated above to correct the errors, and Applicant submits that this objection has been overcome.

Section 102(b) Rejections:

The Examiner rejected claims 1, 3-5, 24, 25 and 27-29 under 35 U.S.C. § 102(b) as being anticipated by Kurihara (U.S. Patent 4,107,649) (hereinafter, "Kurihara"), and claim 30 under 35 U.S.C. § 102(b) as being anticipated by Arroyo et al. (U.S. Patent 5,502,732) (hereinafter, "Arroyo"). Applicant respectfully traverses these rejections and submits that neither Kurihara nor Arroyo anticipates claims 1, 3-5, 24, 25 and 27-30, as set forth in detail below.

Regarding claim 1, contrary to the Examiner's assertion, Kurihara does not disclose shifting a first bit having a different logical value than the same logical value across the initial data bit combination, wherein each time the first bit is shifted, one of n data bit combinations is generated. In contrast, Kurihara teaches a system in which a shift register sequentially shifts input data and feeds back output data and in which a parity signal is generated and compared to a predicted, or counted, parity signal to determine whether the shift register is working correctly. Specifically, Kurihara teaches that input data is shifted through a shift register while the number of logical values of '1' in the input data are counted and the number of logical values of '1' outputted by the shift registers is also counted (Kurihara, column 3, lines 13-24). Additionally, Kurihara's shift register performs a parity calculation on the data in the shift register. Kurihara then uses the number of 1s counted in the input data and/or the number of 1s counted in the output data to predict the parity value for the data in the shift register (Kurihara, column 3, lines

25-38). Kurihara's system includes a check circuit to compare the predicted or counted parity value with the parity value generated by the shift register and generates an error signal if they do not match (Kurihara, column 4, lines 13-19). However, Kurihara does not teach shifting a first bit having a different logical value across the initial data bit combination, wherein each time the bit is shifted, one of n data bit combinations is generated and provided to error detection/correction logic. While Kurihara does use a shift register and does shift input data through the shift register, Kurihara does not teach generating one of n data bit combinations each time the first bit is shifted and providing each of the bit combinations to error detection/correction logic. Kurihara only teaches that input data is shifted in to the shift register.

In the "Response to Arguments" section of the Final Action, the Examiner disagrees with the foregoing and asserts that "Kurihara teaches an error detection circuit which comprises a shift register... which sequentially shifts input data... [and that] the output from each stage of the shift register 101 is supplied to the parity generator 4, and a parity signal PARITY 1 is derived therefrom...." In response, Applicant asserts that Kurihara does not teach or suggest the specific configuration of these structures that is required by Applicant's claim. Applicant's claim specifically recites the steps of creating a data bit combination such that each data bit of an initial data bit combination has a same logical value as each other bit, and shifting a bit having a different logical value than the same logical value across the bit combination. Whether Kurihara's structure could or could not hypothetically be configured to perform these steps in hindsight is irrelevant to the question of anticipation of Applicant's method claim. As noted below, anticipation requires that Kurihara disclose the steps recited in Applicant's claim, arranged as in the claim, in as complete detail as recited in the claim. Kurihara simply fails to do so.

In further regard to claim 1, Kurihara fails to teach providing each of the n data bit combinations to the error detection/correction logic; in response to said providing, the error detection/correction logic generating a set of check bits for each of the n data bit combinations; comparing the set of check bits generated by the error correction/detection logic with a known correct set of check bits for each of the n data bit combinations.

Instead, Kurihara teaches only that input data enters the input end of the shift register (Kurihara, column 3, lines 13-15). Kurihara mentions nothing about providing each of n data bit combinations to the error detection/correction logic. Kurihara does not mention anything about the size of input data nor about the input data including n data bit combinations. Kurihara is equally silent regarding the error detection/correction logic generating a set of check bits. Instead, Kurihara's shift register calculates the single parity bit from the input data in the shift register. Calculating a single parity bit is very different from generating a *set of check bits*. Kurihara also does not disclose comparing the set of check bits generated by the error correction/detection logic with a known correct set of check bits for each for the n data bit combinations. In contrast, Kurihara compares a generated parity bit with a predicted, or counted, parity bit from the input data (and/or output data). The counted parity value of Kurihara is clearly not a *known correct set of check bits*. Furthermore, Kurihara does not teach comparing the set of check bits with a known correct set of check bits *for each of the n data bit combinations*. Kurihara only teaches that input data is shifted into the shift register and that the number of logical values of 1 in the input data are counted. Kurihara does not mention n data bit combinations nor does he disclose comparing check bits with known correct check bits *for each of the n data bit combinations*.

In the "Response to Arguments" section of the Final Action, the Examiner disagrees with the foregoing and asserts that Kurihara teaches comparing a predicted parity value with the parity value produced by the error detection circuit. However, as argued above, generation of a single parity bit is not in any way equivalent to generating a set of check bits, as recited in Applicant's claim. Further, Kurihara does not teach a comparison against a known correct set of check bits. Kurihara's predicted parity bit is not a known correct parity bit. Rather, it is simply a parity bit generated through redundant means independent of the main error detection circuit. The fact that a parity bit is generated redundantly does not entail that the resulting parity bit is known to be correct. Kurihara again fails to meet the standard of anticipation with respect to these features.

For at least the foregoing reasons, Applicant submits that Kurihara fails to anticipate claim 1.

Regarding claim 24, Kurihara fails to teach test check bit generating means for creating a set of test data bit combinations and providing the set of test data bit combinations to a check bit generator comprised in the error correction/detection logic. In contrast, as discussed above, Kurihara teaches inputting data to a shift register. Kurihara does not disclose creating a set of test data bit combinations and providing the set of test data bit combinations to a check bit generator. The Examiner cites FIG. 2 and column 5, lines 7-34 of Kurihara. The cited portions of Kurihara only describe shifting input data into a shift register, generating a parity value for the input data in the shift register and counting the values of 1 in the input data for comparison. The Examiner appears to be implying that Kurihara's input data is a set of test data bit combinations and that Kurihara's shift register is a check bit generator. However, Kurihara does not teach that the input data to the shift register is a set of test data bit combinations created by test check bit generating means.

In the "Response to Arguments" section of the Final Action, the Examiner disagrees with the foregoing and asserts that Kurihara teaches an error detection circuit including a shift register for receiving input data and an output stage for feeding back output data, as well as a parity generator for generating a parity signal from the data stored in the shift register. However, as argued above, Kurihara teaches only that a parity bit may be generated in several redundant ways (parity generator 4, counters 2 and 3) for a given set of data bits resident within a shift register. Kurihara does not suggest in any way that the data bits resident within the shift register are test data bit combinations created by test check bit generating means. In fact, Kurihara is entirely silent as to the source of the input data stored within the shift register.

Further regarding claim 24, Kurihara also fails to disclose wherein the set of test data bit combinations comprises n n-bit data bit combinations, wherein each possible value of each data bit is present in at least one of the n n-bit data bit combinations in the

set of test data bit combinations. Nowhere does Kurihara mention anything about a set of test data bit combinations comprising n n-bit data bit combinations and the Examiner's cited passages (FIG 2 and column 5, lines 7-34) only describe shifting input data into a shift register, generating a parity value for the data in the shift register and counting the values of 1 in the input data for comparison. Neither the cited portions, nor the remainder of Kurihara, mention anything regarding n n-bit data combinations. Furthermore, nowhere does Kurihara teach that each possible value of each data bit is present in at least one of the n n-bit data bit combinations of test data bit combinations. Kurihara only refers generically to "input data" without describing anything about any combinations of bits in the makeup of the input data (see, Kurihara, column 1, lines 44-48, and column 3, lines 13-17).

Additionally, Kurihara fails to teach comparison means for comparing check bits output by the error/detection logic for each of the n n-bit data bit combinations in the set of test data bit combinations to known correct check bits for each of the n n-bit data bit combinations. Kurihara only teaches comparing a one-bit predicted parity value, generated by counting the "1"s in input data, with a one-bit parity value generated on the data in the shift register. Nowhere does Kurihara disclose anything regarding comparing check bits for each of n n-bit data bit combinations. As noted above, Kurihara fails to mention n n-bit data bit combinations at all.

In the "Response to Arguments" section of the Final Action, the Examiner disagrees with the foregoing and asserts that Kurihara teaches a check circuit configured to compare the predicted parity value with the actual parity value. However, as argued above with respect to claim 1, Kurihara does not teach the generation of check bits corresponding to each of several data bit combinations and comparing the generated check bits against known correct check bits. Further, as argued above, Kurihara fails to teach or suggest that the n-bit data bit combinations are included as part of a set of test bit data combinations.

Applicants respectfully remind the Examiner that Anticipation requires the presence in a single prior art reference disclosure of each and every limitation of the claimed invention, arranged as in the claim. *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The identical invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). Kurihara clearly fails to meet this standard with respect to claim 24, and so cannot anticipate claim 24.

Regarding claim 27, in contrast to the Examiner's contention, Kurihara does not teach a method including providing a set of m+1 test code words to the error correction/detection logic, wherein each code word has m bits, wherein a first test code word in the set of m+1 test code words is a correct code word, wherein each test code word other than the first test code word comprises a single-bit error at a different bit position within the code word than each other test code word. The Examiner's cited portions (FIG 2, column 2, lines 61-68 and column 3, lines 13-34) only describe how Kurihara calculates a predicted parity value and compares it to a generated parity value to detect malfunction of the error detection circuit. However, nowhere in the cited passages does Kurihara mention providing a set of m+1 test code words to the error correction/detection logic. The Examiner is apparently equates Kurihara's references to "input data" shifted into a shift register as teaching providing a set of m+1 test code words to the error correction/detection logic. However, Kurihara never mentions a set of m+1 test code words, nor does he describe his "input data" as test code words. Further, Kurihara does not describe his input data as having m bits. In fact, Kurihara never mentions anything about the size of input data. Thus, Kurihara's input data cannot be equated to a set of m+1 test code words, wherein each code word has m bits, as the Examiner contends.

Additionally, Kurihara fails to disclose wherein a first test code word in the set of m+1 test code words is a correct code word, wherein each test code word other than the first test code word comprises a single-bit error at a different bit position within the code word than each other test code word. Kurihara's method simply does not include a set of

m+1 test code words that includes a first test code word that is a correct code words and wherein the other test code words each include a single-bit error at different bit positions. Kurihara teaches a completely different type of method incompatible with a set of test code words. Kurihara's entire method is only concerned with comparing predicted and generated parity values. It would not even make sense to use a set of m+1 test code words where every code word other than a first correct code word includes single bit errors in Kurihara's method, because his method would merely compare the predicted parity value (total counted "1"s) with a generated parity value (by a parity generator). Such a method has no way to detect or make use of single-bit errors in test code words.

Furthermore, Kurihara's method clearly does not include the error correction/detection logic decoding the set of m+1 test code words. As noted above, Kurihara does not teach providing a set of m+1 test code words to the error correction/detection logic. Kurihara also fails to teach error correction/detection logic that decodes such a set of m+1 test code words. Kurihara's entire system includes only a shift register, 2 one-bit counters, a parity generator, and a check circuit (FIGs 1, 2, and column 3, lines 39-65). There is nothing in Kurihara's system that is described as capable of decoding a set of m+1 test code words.

In the "Response to Arguments" section of the Final Action, the Examiner disagrees with the foregoing and asserts that Kurihara teaches an error detection circuit including a shift register that sequentially shifts input data and feeds back output data to desired shift register stages, as well as a parity generator that generates a parity signal in accordance with data stored in the shift register. The Examiner further asserts that "the input data and the shift register provides the test code words to the error correction/detection logic." Applicant disagrees with the Examiner's interpretation of Kurihara. As argued above, Applicant's claim 27 specifically recites test code words having definite characteristics, such as the positioning of single-bit errors within the test code word. Kurihara does not disclose any aspect of specifically positioning single-bit errors within a set of test code words. In fact, as argued above with respect to claim 24, Kurihara does not teach that the input to the shift register is configured in any deliberate

way for testing of error detection logic or for other purposes. Rather, Kurihara merely computes parity in two different ways and assumes that if the two sources of parity disagree, an error exists. Kurihara makes no teaching whatsoever regarding deliberately introducing errors into any portion of the shift register or parity generation logic, as recited in claim 27.

For at least the foregoing reasons, Kurihara fails to anticipate claim 27.

Regarding claim 30, contrary to the Examiner's assertion, Arroyo does not teach a method including providing a set of test code words to the error correction/detection logic, wherein said providing comprises introducing an error into each of the test code words in the set by substituting check bits corresponding to an unused syndrome for a correct set of check bits within each test code word, wherein each test code word comprises substituted check bits corresponding to a different unused syndrome than each other test code word in the set of test code words. Arroyo teaches a system for checking the test logic contained in a computer memory system during POST (Abstract). The Examiner's cited passage (Arroyo, column 5, line 55-column 6, line 4) describes how a CPU writes data including two bits set to "1" to memory and how a diagnostic bit controlling multiplexer 33 replaces the two bits set to "1" with zeros. In Arroyo's system ECC generator 31 generates check bits for the unmodified data and stores those check bits in memory. When the data is read from memory (including the zeros substituted by multiplexer 33), ECC generator 41 generates check bits that does not match the check bits originally generated by ECC generator 31 when the data was stored. Hence, Arroyo does not substitute check bits corresponding to an unused syndrome, but rather teaches replacing two bits set to "1" with bits set to "0". These are two very different techniques.

Further, Arroyo's method does not include providing a set of test code words, wherein each test code word comprises substituted check bits corresponding to a different unused syndrome. In contrast, Arroyo teaches that "a multiplexer is provided in the data write path which substitutes *a constant set of identical bits* for the actual data generated by the CPU" (emphasis added, Arroyo, column 2, lines 22-25). Arroyo clearly teaches

substituting a constant set of identical bits and thus clearly teaches away from providing a set test code words wherein each test code word comprises substituted check bits corresponding to a different unused syndrome.

In the "Response to Arguments" section of the Final Action, the Examiner disagrees with the foregoing and cites Arroyo's description of testing ECC logic at col. 7, lines 22-44. However, Applicant notes that as argued above, Arroyo does not teach introducing an error into a code word by substituting check bits corresponding to an unused syndrome for a correct set of check bits in the code word, as recited in claim 30. Rather, Arroyo introduces an error into a code word simply by zeroing out the check bits read from memory. The resulting syndrome of Arroyo is thus entirely dependent upon the check bits computed from the data read from memory, since the syndrome is formed from the logical XOR of the computed check bits and the check bits read from memory, but a logical XOR of any value with the zero is simply the value itself. Since Arroyo is substituting only zeroes for the check bits read from memory, Arroyo simply cannot function to deliberately substitute check bits corresponding to an unused syndrome, as required by Applicant's claim 30. Instead, whether a resulting syndrome of Arroyo is unused depends upon the data read from memory. But Arroyo does not teach any substitution of even the data bits in this case: the data written to memory at step 4 of FIG. 5a is the same as the data read from memory at step 5 of FIG. 5b, which is then used to generate the check bits compared with the check bits that have been forced to zero. Thus, Arroyo does not teach any aspect of substituting either check bits or data bits corresponding to an unused syndrome. As argued previously, the technique of Arroyo is fundamentally dissimilar from the technique recited in claim 30.

For at least the reasons given above, Arroyo cannot be said to anticipate claim 30, and therefore Applicant submits that claim 30 is distinguishable.

Section 103(a) Rejections:

The Office Action rejected claim 2 under 35 U.S.C. § 103(a) as being unpatentable over Kurihara in view of Nielson et al. (U.S. Patent 5,619,642) (hereinafter, "Nielson"), claims 6-12, 26, 31-33 and 35 as being unpatentable over Kurihara as applied to claim 3 above and further in view of Arroyo, claims 13-16 as being unpatentable over Kurihara in view of Fielder et al. (U.S. Patent 6,446,037) (hereinafter, "Fielder"), claims 17-23 as being unpatentable over Kurihara and Fielder and further in view of Arroyo, and claim 34 as being unpatentable over Kurihara in view of Vishlitzky et al. (U.S. Patent 5,809,332) (hereinafter, "Vishlitzky").

Regarding claim 34, Kurihara in view of Vishlitzky fails to teach providing each of a set of n data bit combinations to the error detection/correction logic. Kurihara mentions nothing about providing each of n data bit combinations to the error detection/correction logic. Kurihara does not mention anything about the size of input data nor does he mention input data including a set of n data bit combinations. The Examiner has not cited any passage of Kurihara that discusses a set of n data bit combinations. Applicant notes that remarks presented above regarding claim 1 apply to claim 34 as well. Specifically, Applicant notes that Kurihara also fails to teach generating a set of check bits, as distinct from the single parity bit of Kurihara, corresponding to an input data bit combination. Kurihara further fails to teach comparing the generated check bits with a known correct set of check bits for each input data bit combination. Also, Vishlitzky fails to teach or suggest those limitations of claim 34 that are omitted by Kurihara.

For at least the reasons given above, Kurihara in view of Vishlitzky fails to teach or suggest all of the limitations of claim 34. Applicant therefore submits that claim 34 is distinguishable.

Regarding claim 35, Kurihara in view of Arroyo fails to teach providing a subset of possible data bit combinations of n data bits to the error detection/correction logic, wherein the subset comprises n data bit combinations, wherein each possible value of each data bit is present in at least one of the n data bit combinations in the subset. As

noted above regarding claim 1, Kurihara teaches comparing a one bit predicted parity value with a one bit generated parity value. Arroyo teaches substituting "a constant set of identical bits [zeros]" (Arroyo, column 2, lines 22-25). Neither Kurihara nor Arroyo teach providing a subset of data bit combinations of n data bits, wherein each possible value of each data bit is present in at least one of the n data bit combinations. Kurihara only teaches input data without mentioning anything regarding possible data bit combinations of n data bits and Arroyo teaches the use of a constant set of identical bits.

Additionally, Kurihara in view of Arroyo also fails to teach verifying the error detection/correction logic by comparing a set of check bits generated by the error detection/correction logic for each of the n data bit combinations in the subset with a set of known correct check bits. The Examiner contends that Kurihara teaches this. However, the Examiner's cited portions of Kurihara (FIG 2, column 1, lines 44-60 and column 2, lines 12-19) only refer to comparing a predicted parity value with a generated parity value for input data in a shift register. Nowhere does Kurihara mention anything about comparing a set of check bits generated by the error detection/correction logic for each of n data bit combinations. In fact, as noted above, Kurihara fails to mention n data bit combinations as all and certainly does not disclose comparing a set of check bits generated for each of n data bit combinations with a set of known correct check bits. Arroyo also fails to teach such functionality. In contrast, as noted above, Arroyo teaches substituting a constant set of identical bits (zeros) when writing data to memory and comparing check bits generated on both the original data and the data with zeros substituted. Since neither Kurihara, nor Arroyo teaches comparing check bits generated for each of n data bit combinations with a set of known correct check bits, no combination of Kurihara and Arroyo would include such a feature.

Kurihara in view of Arroyo also fails to teach providing a first set of m+1 test code words to the error detection/correction logic, wherein a first test code word is a correct test code word and where each other test code word in the set of m+1 test code words comprises a single-bit error, wherein each test code word having a single-bit error has the single-bit error at a different bit position than each other test code word that has a

single-bit error. The Examiner asserts that Kurihara teaches such functionality in his method and cites Fig 2, column 2, lines 61-68 and column 3 lines 13-34. However, as noted above regarding the rejection of claim 27, these cited portions fail to mention anything about a set of m+1 test code words and the remarks and arguments presented above regarding claim 27 also apply here. Arroyo also fails to teach providing a set of m+1 test code words wherein a first test code word is a correct test code word and where each other test code word comprises a single bit error at a different bit offset than each other test code word. As noted above, Arroyo teaches that "a multiplexer is provided in the data write path which substitutes *a constant set of identical bits* for the actual data generated by the CPU" (emphasis added, Arroyo, column 2, lines 22-25). Arroyo further teaches an ECC generator 31 that generates check bits for the unmodified data and stores those check bits in memory. When the data is read from memory (including the zeros substituted by multiplexer 33), ECC generator 41 generates check bits that does not match the check bits originally generated by ECC generator 31 when the data was stored. Hence, Arroyo does not provide a set of m+1 test code words, but rather teaches comparing check bits generated for a set of data with check bits generated on the same data after having zeros inserted for certain bits. Thus, Arroyo clearly fails to teach providing a set of m+1 test code words wherein a first test code word is a correct test code word and where each other test code word comprises a single bit error at a different bit offset than each other test code word. Additionally, no combination of Kurihara and Arroyo can include such a feature.

Kurihara in view of Arroyo also fails to teach providing a second set of test code words to the error detection/correction logic, wherein each test code word in the second set comprises an error introduced by substituting check bits corresponding to an unused syndrome for a correct set of check bits within a correct code word, wherein each test code word in the second set comprises substituted check bits corresponding to a different unused syndrome than each other test code word in the second set of test code words. Kurihara clearly fails to teach substituting check bits corresponding to an unused syndrome and the Examiner relies upon Arroyo for this feature. However, Arroyo also fails to teach substituting check bits corresponding to an unused syndrome. Please see

above regarding claim 30 for remarks regarding how Arroyo fails to teach substituting check bits corresponding to an unused syndrome.

In the "Response to Arguments" section of the Final Action, the Examiner disagrees with the foregoing and reiterates numerous aspects of Kurihara and Arroyo discussed above with respect to other claims. Applicant makes specific reference to the arguments set forth above with respect to claims 27 and 30, each of which include limitations also present in claim 35. As argued above, Kurihara and Arroyo fail to anticipate these respective claims, and neither reference suggests the limitations that the other fails to teach. Therefore, the combination of Kurihara and Arroyo cannot teach or suggest the combination recited in claim 35. Consequently, Applicant submits that claim 35 is distinguishable.

Regarding claim 13, Kurihara in view of Fielder fails to teach a computer readable medium comprising program instructions computer executable to: create an initial data bit combination having n bits, wherein each data bit in the initial data bit combination has a same logical value as each other data bit in the initial data bit combination; shift a first bit having an different logical value than the same logical value across the initial data bit combination, wherein each time the first bit is shifted, one of n data bit combinations is generated; provide each of the n data bit combinations to error detection/correction logic; compare a set of check bits generated by the error correction/detection logic with a known correct set of check bits for each of the n data bit combinations; and dependent on an outcome of said comparing, generate an indication of whether the error detection/correction logic correctly generated the set of check bits. Please note that the remarks above regarding claim 1 in view of Kurihara also apply here to claim 13.

Additionally, Fielder teaches a method for scalable coding of audio data into a core layer in response to a desired noise spectrum established according to psychoacoustic principles that supports coding augmentation data into augmentation layers in response to various criteria (Fielder, Abstract). Fielder's method has nothing to

do with the error detection/correction circuit of Kurihara. The Examiner is relying upon Fielder to teach a computer readable medium. However, the respective inventions of Kurihara and Fielder are directed to completely different fields of endeavor. Applicant can imagine no conceivable way to combine Kurihara's error detection circuit checking circuit with Fielder's scalable audio encoding process. Kurihara teaches a hardware circuit for detecting malfunction of an error detection circuit and does not teach anything related to Fielder's scalable audio encoding software. The Examiner's cited passage from Fielder (column 4, lines 60-64) describes how Fielder's scalable audio encoding and decoding processes may be conveyed by a machine readable medium. However, Kurihara's teachings are specific to a hardware circuit implementation. There is no suggestion to in either Fielder or Kurihara to create a program instruction computer readable medium implementation of Kurihara's circuit. In fact, since Kurihara's teachings are specific to a hardware circuit, it is unclear how Kurihara's teachings could even be applied to a program instruction computer readable medium implementation.

Moreover, the Fielder reference is not analogous art. "In order to rely on a reference as a basis for rejection of an applicant's invention, the reference must either be in the field of applicant's endeavor or, if not, then be reasonably pertinent to the particular problem with which the inventor was concerned." *In re Oeticker*, 977 F.2d 1443, 1446, 24 USPQ2d 1443, 1445 (Fed. Cir. 1992). "A reference is reasonably pertinent if, even though it may be in a different field from that of the inventor's endeavor, it is one which, because of the matter with which it deals, logically would have commended itself to an inventor's attention in considering his problem." *In re Clay*, 966 F.2d 656, 659, 23 USPQ2d 1058, 1060-61 (Fed. Cir. 1992). Here, Fielder is clearly not in the field of Applicant's endeavor of testing error correction/detection logic. In contrast, Fielder deals with scalable encoding and decoding of audio data (Fielder, col. 2, lines 35-49). Furthermore, the subject of Fielder would not logically have commended itself to an inventor's attention when considering the problem addressed by Applicant. One of skill in the art seeking to address the problem of testing error correction/detection logic would not have any logical reasons for considering a technique used to encode and decode audio data. Thus, Fielder is not within Applicant's field of endeavor and is not

pertinent to the problem addressed by Applicant's invention. Not is Fielder within the field of endeavor or pertinent to the problem addressed by Kurihara. Accordingly, Fielder is non-analogous art and cannot properly be combined with Kurihara.

The Examiner states that Fielder is in an analogous art. However, the Examiner has clearly over-generalized the meaning of "analogous art." *In re Oeticker* refers to Applicant's field of endeavor and particular problem. The analogous art requirement can always be made meaningless by over-generalizing the field of endeavor or problem. Almost any art may be considered pertinent if the problem is stated in general enough terms. That is why the courts have insisted that art used in § 103 rejections be in the same field of endeavor or pertinent to the particular problem. Fielder pertains to scalable encoding and decoding of audio data, which has nothing to do with Applicant's field of endeavor or particular problem.

In the "Response to Arguments" section of the Final Action, the Examiner disagrees with the foregoing and reasserts that Fielder teaches a computer readable medium. However, the Examiner's assertions do nothing to address Applicant's contention that Fielder is not analogous art with respect to Applicant's claimed invention.. As argued above, the problem addressed by Fielder, that of audio encoding, has nothing whatsoever to do with the problem of testing an error detection/correction system. Applicant maintains that one of skill in the art in testing error detection systems would have no motivation whatsoever to consult art relating to audio encoding methods. Moreover, Applicant refers to the additional arguments made above with respect to claim 1 in support of the contention that Kurihara fails to teach or suggest similar limitations of claim 13.

Therefore, for at least the reasons given above, the rejection of claim 13 is not supported by the prior art, and Applicant submits that claim 13 is distinguishable.

Applicant notes that many of the dependent claims recite further distinctions unsupported by the cited art. However, as the independent claims have been shown to be distinguishable, further discussion of the dependent claims is unnecessary at this time.

CONCLUSION

Applicant submits the application is in condition for allowance, and notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-94400/RCK.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard

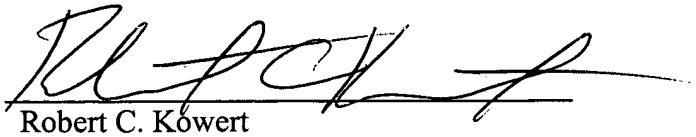
☐ Petition for Extension of Time

☐ Notice of Change of Address

☐ Fee Authorization Form authorizing a deposit account debit in the amount of $ for fees ( ).

☐ Other:

Respectfully submitted,



Robert C. Kowert
Reg. No. 39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: May 23, 2005